Enhancing Interaction with Smart Objects through Mobile Devices

Ana M. Bernardos, José R. Casar, Jesús Cano, Luca Bergesio
Universidad Politécnica de Madrid
abernardos, jramon, jcano, luca.bergesio}@grpss.ssr.upm.es

ABSTRACT
Interaction with smart objects can be accomplished with different technologies, such as tangible interfaces or touch computing, among others. Some of them require the object to be especially designed to be 'smart', and some other are limited in the variety and complexity of the possible actions. This paper describes a user-smart object interaction model and prototype based on the well known event-condition-action (ECA) reasoning, which can work, to a degree, independently of the intelligence embedded into the smart object. It has been designed for mobile devices to act as mediators between users and smart objects and provides an intuitive means for personalization of object's behavior. When the user is close to an object, this one publishes its 'event & action' capabilities to the user's device. The user may accept the object's module offering, which will enable him to configure and control that object, but also its actions with respect to other elements of the environment or the virtual world. The modular ECA interaction model facilitates the integration of different types of objects in a smart space, giving the user full control of their capabilities and facilitating creative mash-uping to build customized functionalities that combine physical and virtual actions.

Categories and Subject Descriptors
H.1.2 User/Machine Systems, H.5.2 User Interfaces.

General Terms
Design, Experimentation, Human Factors.

Keywords
Smart objects, smart environments, interaction, mobile middleware.

1. INTRODUCTION
A number of works have proposed sound technical solutions to interact with smart objects through personal devices. In 1999, Want et al. [1] explored how to link objects to digital resources through RFID tags, and Biegl [2] implemented the *point&click* concept in a generic device: when pointed at other devices for selection, the laser-based 'remote control' was capable of obtaining their control information, allowing operational control with the help of a simple user interface; when the object detects the laser beam, it transfers the control description to the device, using infrared.

These touching and pointing paradigms for physical interaction between devices, and others (e.g. scanning and user-mediated object interaction [3]), have been used to implement the *smart space* concept, by exploiting mobile personal devices capabilities. In [4], smart objects use formatted SMS containing simple commands to enable their remote operation though a mobile device. A framework for mobile devices to interact with NFC tagged physical objects that are associated with web services and provide information for their invocation is described in [5].

Smart objects are the elements that may make a space smart enough to adapt to users' preferences or needs. The concept of 'smart object' may include any type of device with sensing and processing capabilities; in this paper, we will use it to refer to *'a computationally augmented tangible object with an established purpose that is aware of its operational situations and capable of providing supplementary services without compromising its original appearance and interaction metaphor'* [6]. Thus, each smart object, independently of its original function, may offer some supplementary services, such as e.g. environmental sensing, information display or user identification.

This paper describes an interaction model and prototype that intends to facilitate the user's control on the supplementary services which can be offered by a smart object. The interaction model is enabled by an architecture that makes possible for smart objects to publish their 'event detection' and 'action' capabilities to the mobile devices nearby. Through their mobile devices, users are able to configure and use the objects' capabilities to build personalized interaction mash-ups. From the user point of view, the key characteristics of the system are the adaptability of object's behavior to user preferences and the simplicity of programming.

The paper is organized as follows. Section 2 explains, from the point of view of the demonstration system objectives, the well known Event-Condition-Action (ECA) interaction concept using a simple functional description of some use cases. Section 3 describes the architectural approach to implement the ECA interaction model. Section 4 briefly completes some details of the demo concept. Section 5 concludes the paper.

2. THE ECA INTERACTION MODEL
Let us consider a space populated with different types of objects, including stand-alone and networked smart objects, with or without embedded processing and decision capabilities. In our model, the mobile phone may enhance the processing, storage, identification, sensing, interface and communication capabilities [4] of these smart objects. Through it, the user will be able to:

MobiWac'11, October 31–November 4, 2011, Miami, Florida, USA.

- Customize the responses of tangible interaction with a given object to induce actions, e.g. 'if I turn the world globe right, switch the light off'; 'if I select a country in the globe, show related travel video contents on the TV set'.
- Make an object respond (mechanically, physically or virtually) to a given order configured by the user. E.g. 'if I touch the knob and my keys are onto a smart furniture, display a message with the key's location'.
- Provide intuitive configuration of the smart environment through actions in the mobile device. E.g. 'if I shake the phone, put the music on'.
- Configure/activate features in the mobile device depending on environmental events. E.g. 'if I switch the light off, the room is silent and it is Sunday midnight, configure my mobile alarm clock in working settings'.

The ECA interaction model aims at providing bidirectional interaction between objects and user (mobile) devices, in order to make possible to configure the object's actions from the mobile device, but also to enable that one given object's events could change other objects actions or even the user device configuration.

The model is based on the following seven design principles:

1. *Proximity detection*: through different technologies and interaction paradigms (touching, scanning…), a mobile device and/or a smart object may detect each other. Proximity will be the starting point for interaction, but it will not be necessary afterwards, since the smart objects and the user device will communicate/interact through a central/common ubiquitous network and server.
2. *Module publishing*: smart objects will be able to publish event and action 'modules'. A 'module' will implement a number of logic interfaces to hide the operational complexity to detect an event or perform an action. An 'event' is any episode of change that may be detected by a sensor either embedded in the object or placed somewhere in the local environment (e.g. changes in temperature, light, pressure, etc.). An 'action' is any task that may be performed by or on the object (e.g. 'display an image', 'change my colour', 'switch the light off'). Modules may be configurable by the user, depending on their function.
3. *Rule-based reasoning*, by including 'condition' modules: an action will be triggered when an event occurs and one or several 'conditions' is/are fulfilled. Default 'condition' modules might be available in the mobile device for some common cases.
4. *ECA bundle configuration*: mobile mash up tools will allow the user to easily configure ECA bundles as a combination of event, conditions and actions. Smart objects will be able to publish aggregated common ECA bundles, to prevent the user from manually configuring them.
5. *Univocal module offering identification*: each module offering will be univocally identified and will be associated to a smart object. The mobile device will be able to access an object's offering by its identifier.
6. *Inter-object interaction*: the actions in or for an object are not necessarily associated to events defined by or in that same object, but rather, possibly associated to events and conditions in other objects in the same (real or virtual) space. This cross-interaction among objects and user is also a key characteristic of our concept of smart space.
7. *Module life-cycle management*: in order to download, install and maintain unused modules, user feedback and relative location are used. For example, the user will be notified when some new modules are accessible, and downloaded modules will be deleted if not configured and the user exits the objects' area of influence.

When a smart object detects an application-enabled mobile phone nearby, it offers to the phone the download of a set of modules that enables the interaction with the object. After accepting to download the modules, the user can configure ECA bundles with them as described before. When an event occurs in or is detected by a sensor associated to a smart object, the corresponding module in the mobile phone receives a notification. The mobile application then checks the conditions in the ECA bundles (if any) and, if satisfied, executes the configured actions, affecting either the same, or some other smart objects or the device itself.

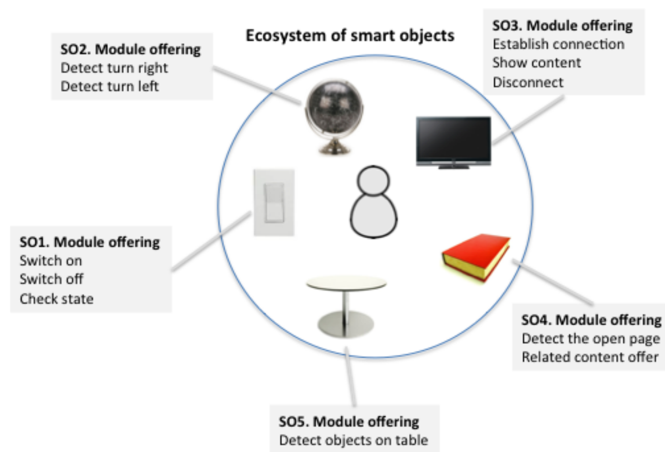

Figure 1. Interaction concept: objects publish their supplementary services offering, which is composed by actions and events. The user is able to manage the components by a simple interface to control and configure the *smart objects* and their interactions with the environment.

It is important to note again that once defined the ECA strategy in an object or among objects in a space, the operation does not require any proximity between the mobile device and the object(s), since the communications and interactions among them are to be managed through a centralized ubiquitous infrastructure and a common server, and not necessarily in a point to point basis (which would be theoretically and practically possible, too).

3. MOBILE BASED-ARCHITECTURE TO ENABLE THE ECA INTERACTION MODEL

When detecting an object for the first time (touch or scan interaction), the mobile device will retrieve an identifying URL to download, from a server, the object's module offering, together with some examples of ECA bundles (Fig. 2 shows the activity diagram of the process). For a practical implementation of the ECA model, smart objects might be equipped with Bluetooth motes or NFC tags, for example.

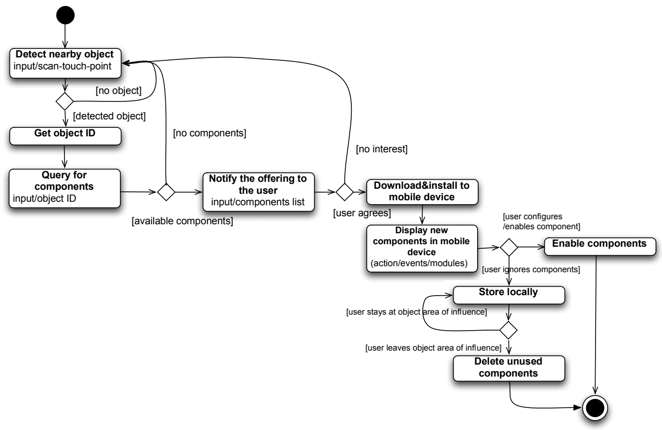

Figure 2. Activity diagram for the interaction model.

The mobile application is the central element to manage interaction. It is divided into three main building blocks: *core*, *application interface* and *modules*. The *core* manages the modules' lifecycle: it dynamically retrieves them from the infrastructure and loads them into memory, manages the interaction between events, conditions and actions and, finally, provides the GUI. The *application interface* defines the data structure for the communication between the core and the modules. Finally, each ECA bundle aggregates *module* events, conditions and actions to interact with a smart object or with the mobile itself. A module offering may include none, one or more than one of each (events, conditions and actions).

For example, Fig. 3 shows the class diagram for the core application module. *SaposCore* is the kernel of the application: it orchestrates the functioning of the whole application, including the graphic user interface and when the application is minimized to run in daemon mode. *Engine* handles the modules available to implement new ECA bundles. *Manager* handles the ECA bundles and the interaction between them and events, conditions and actions supported by the other classes (*Workers* and *Player*). *SapoDef* is a data class, it permits storing ECA bundles and access their information.

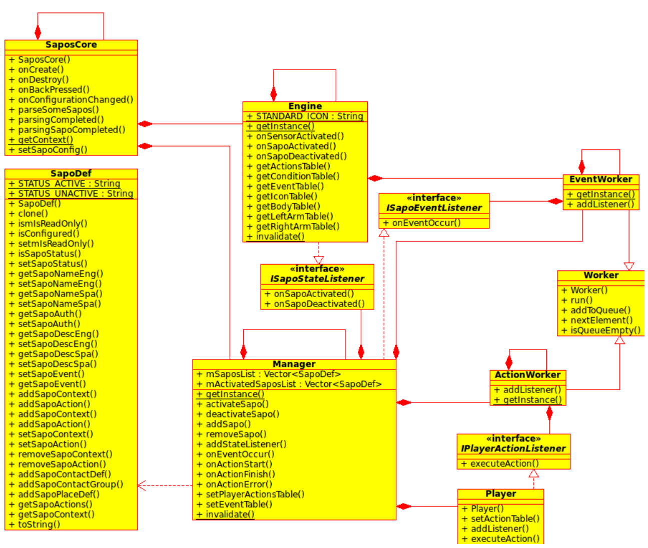

Figure 3. Class diagram for the core module of the ECA model enabling mobile application.

The approach in this paper assumes that the user is able to explicitly operate the system, being fully aware of the tool s/he is using. For this reason, the mobile interface is very simple and intuitive allowing straight component management and ECA module composition. Developing new modules wrapping objects' functionalities is also straight, due to the definition of APIs for integration in the mobile architecture.

4. PROTOTYPE AND DEMO

Our current prototype for the ECA model in a smart space has been implemented on Android v2.3, in a Google Nexus S smartphone, which includes NFC technology.

The prototype implements the ECA interaction model through the described mobile application that enables *module* life cycle management (to dynamically download, install, configure, combine or delete *Event*, *Condition* and *Actions* software units) and provides a very simple interface for the user to deal with ECA bundles configuration. The mobile application provides a standard set of events, conditions and actions to manage the mobile device itself (control sensors, interfaces and communication capabilities in it), and also to manage, under the ECA philosophy, a set of smart objects.

In the demo, some objects will be real and domotically controlled and some other will be emulated on Android-based tablets. In particular, the following functionalities are demonstrated: how to customize the responses of tangible interaction with a given object to induce actions, how to make an object respond (mechanically, physically or virtually) to a given order configured by the user, how to provide intuitive configuration of the smart environment through actions in the mobile device and how to configure/activate features in the mobile device depending on environmental events.

The objects in the cases of use have been prototyped with NFC and/or Bluetooth components. The system allows users (including occasional visitors) to build new ECA bundles with the available modules to implement their own interaction concepts.

Finally, wrapping of smart object events, conditions and actions in ECA *modules* will be possible. Visitors will be able to explore the APIs themselves, and experience the design and implementation life cycle.

5. CONCLUSIONS

The largely known event-condition-action (ECA) rule-based interaction model is, in this contribution, focused on setting up an user-oriented system to personalize the behavior of a set of smart objects to adapt it to each user's needs or preferences. The ECA interaction approach makes feasible for the user to easily take advantage of the smart objects capabilities, integrating them with personal devices.

The approach allows dealing with different types of smart objects, independently of their different functional paradigms and capabilities. To achieve this objective, it is necessary to set up different interfaces and handlers for different communication technologies.

Additionally, the modular design of the managing application facilitates the design and integration of new control modules.

Our next steps to advance this work are directed to 1) extending the object-to-device direct module offering through mobile embedded servers and 2) designing a context-aware life-cycle module management to prevent the user from dealing with too many service modules.

6. ACKNOWLEDGMENTS
This work is being supported by the Government of Madrid under grant S2009/TIC-1485 and the Spanish Ministry for Science and Innovation under grant TIN2008-06742-C02-01.

The authors also acknowledge fruitful related discussions on the ECA paradigm on mobile devices with engineers and consultants from Deimos-Space Co., under the THOFU initiative, financed by the Spanish Center for the Development of Technology (CDTI).

7. REFERENCES
[1] Want, R., Fishkin, K.P., Gujar, A., Harrison, B.L. 1999. Bridging Physical and Virtual Worlds with Electronic Tags. *Procs. of the Conf. Human Factors in Computing Systems*, ACM Press, pp. 370-377.

[2] Beigl, M. 1999. Point & Click – Interaction in Smart Environments. *Procs. of the First Int. Symposium on Handheld and Ubiquitous Computing*, LNCS 1707, Springer-Verlag, pp. 311-313.

[3] Rukzio, E., Broll, G., Leichtenstern, K., Schmidt, A. 2007. Mobile Interaction with the Real World: An Evaluation & Comparison of Physical Mobile Interaction Techniques. *Procs. of AmI 2007*, LNCS 4794, pp. 1-18.

[4] Siegemund, F., Floerkemeier, C., Vogt, H. 2004. The value of handhelds in smart environments. *Procs. of ARCS*, LNCS 2981, pp. 291-308.

[5] Broll, G., Paolucci, M., Wagner, M., Rukzio, E., Schmidt, A., Hubmann, H. 2009. Perci: Pervasive Service Interaction with the Internet of Things. *IEEE Internet Computing*, vol. 13, no. 6, pp. 74-81.

[6] Kawsar, F., Fujinami, K., Nakajima, T. 2008. Prottoy Middleware Platform for Smart Object Systems. *International Journal of Smart Home*, vol. 2, no. 3.